**Design Of REST API**

**Library Catalogue**

Designing a REST API for managing books, authors, and categories within a library catalogue system.

**Entities:**

1. **Book:** Represents a book in the library catalogue. Attributes may include title, author(s), publication date, ISBN, category, etc.
2. **Author:** Represents an author of a book. Attributes may include name, biography, etc.
3. **Category:** Represents a category or genre of books. Attributes may include name, description, etc.

**Operations:** (The description of entities and operations.)

**Books:**

* **Retrieve all books:** Allows clients to fetch a list of all books in the library catalogue.
* **Retrieve a specific book by ID:** Enables clients to fetch details of a particular book using its unique identifier.
* **Create a new book:** Allows clients to add a new book to the library catalogue.
* **Update an existing book:** Enables clients to modify the details of an existing book.
* **Delete a book:** Allows clients to remove a book from the library catalogue.

**Authors:**

* **Retrieve all authors:** Allows clients to fetch a list of all authors in the library catalogue.
* **Retrieve a specific author by ID:** Enables clients to fetch details of a particular author using their unique identifier.
* **Create a new author:** Allows clients to add a new author to the library catalogue.
* **Update an existing author:** Enables clients to modify the details of an existing author.
* **Delete an author:** Allows clients to remove an author from the library catalogue.

**Categories:**

* **Retrieve all categories:** Allows clients to fetch a list of all categories in the library catalogue.
* **Retrieve a specific category by ID:** Enables clients to fetch details of a particular category using its unique identifier.
* **Create a new category:** Allows clients to add a new category to the library catalogue.
* **Update an existing category:** Enables clients to modify the details of an existing category.
* **Delete a category:** Allows clients to remove a category from the library catalogue.

**REST API DESIGN AND DESCRIPTION:**

The REST API provides endpoints for interacting with the library catalogue system. The base URL for accessing the API is /api/v1/library.

**Books:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Request Method** | **Resource Path or URI** | **Description** | **Query String** | **Returns** | **RCs** |
| **GET** | {Books} | Get an array of books of the book type. | Optional:  Columns=[CSV]  Where = [key]=[CSV]  Orderby=[CSV] | Array of JSON objects. When the columns parameter is present, an array of JSON maps. | 200 OK: Returns a list of books.  404 Not Found: If the requested resource does not exist. |
| **GET** | {Books}/{ID} | Get specific resource matching the ID | Optional:  Columns=[CSV] | JSON object representing the book. | 200 OK  404 Not Found |
| **POST** | {Books} | Create a new book. | JSON Object | Location URI of new resource in HTTP header | 201 Created: If the book is successfully created.  400 Bad Request: If the request body is invalid |
| **PUT** | {Books}/{ID} | Update an existing book. | JSON Object | Location URI of updated resource in HTTP header | 200 OK: If the book is successfully updated.  404 Not Found: If the book with the specified ID does not exist.  400 Bad Request: |
| **DELETE** | {Books}/{ID} | Delete specific book matching the ID. |  |  | 204 No Content: If the book is successfully deleted.  404 Not Found: If the book with the specified ID does not exist. |

**Authors:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Request Method** | **Resource Path or URI** | **Description** | **Query String** | **Returns** | **RCs** |
| **GET** | {authors} | Get an array of authors of the author type. | Optional:  Columns=[CSV]  Where = [key]=[CSV]  Orderby=[CSV] | Array of JSON objects. When the columns parameter is present, an array of JSON maps. | 200 OK: Returns a list of authors.  404 Not Found: If the requested resource does not exist. |
| **GET** | {authors}/{ID} | Get specific resource matching the ID | Optional:  Columns=[CSV] | JSON object representing the book. | 200 OK  404 Not Found |
| **POST** | {authors} | Create a new author. | JSON Object | Location URI of new resource in HTTP header | 201 Created: If the author is successfully created.  400 Bad Request: If the request body is invalid |
| **PUT** | {authors}/{ID} | Update an existing author. | JSON Object | Location URI of updated resource in HTTP header | 200 OK: If the book is successfully updated.  404 Not Found: If the book with the specified ID does not exist.  400 Bad Request: |
| **DELETE** | {authors}/{ID} | Delete specific author matching the ID. |  |  | 204 No Content: If the author is successfully deleted.  404 Not Found: If the author with the specified ID does not exist. |

**Category:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Request Method** | **Resource Path or URI** | **Description** | **Query String** | **Returns** | **RCs** |
| **GET** | {categories} | Get an array of categories of the category type. | Optional:  Columns=[CSV]  Where = [key]=[CSV]  Orderby=[CSV] | Array of JSON objects. When the columns parameter is present, an array of JSON maps. | 200 OK: Returns a list of categories.  404 Not Found: If the requested resource does not exist. |
| **GET** | {categories}/{ID} | Get specific resource matching the ID | Optional:  Columns=[CSV] | JSON object representing the book. | 200 OK  404 Not Found |
| **POST** | {categories} | Create a new category. | JSON Object | Location URI of new resource in HTTP header | 201 Created: If the category is successfully created.  400 Bad Request: If the request body is invalid |
| **PUT** | {categories}/{ID} | Update an existing author. | JSON Object | Location URI of updated resource in HTTP header | 200 OK: If the category is successfully updated.  404 Not Found: If the category with the specified ID does not exist.  400 Bad Request: |
| **DELETE** | {categories}/{ID} | Delete specific category matching the ID. |  |  | 204 No Content: If the author is successfully deleted.  404 Not Found: If the category with the specified ID does not exist. |